الفصل الأول

تمثيل البيانات

- كيف تجرى العمليات الحسابية فى الـ Visual Basic.Net ؟

١- يتم استقبال البيانات بواسطة الأداة TextBox .
٢- تستقبل أداة الـ TextBox البيانات على أساس أنها بيانات حرفية .
٣- يتم تحول البيانات من النوع الحرفي إلى بيانات من النوع الرقمي .
٤- تجرى العمليات الحسابية داخل ذاكرة الحاسب .
٥- تعرض النتائج بالتنسيق المطلوب .

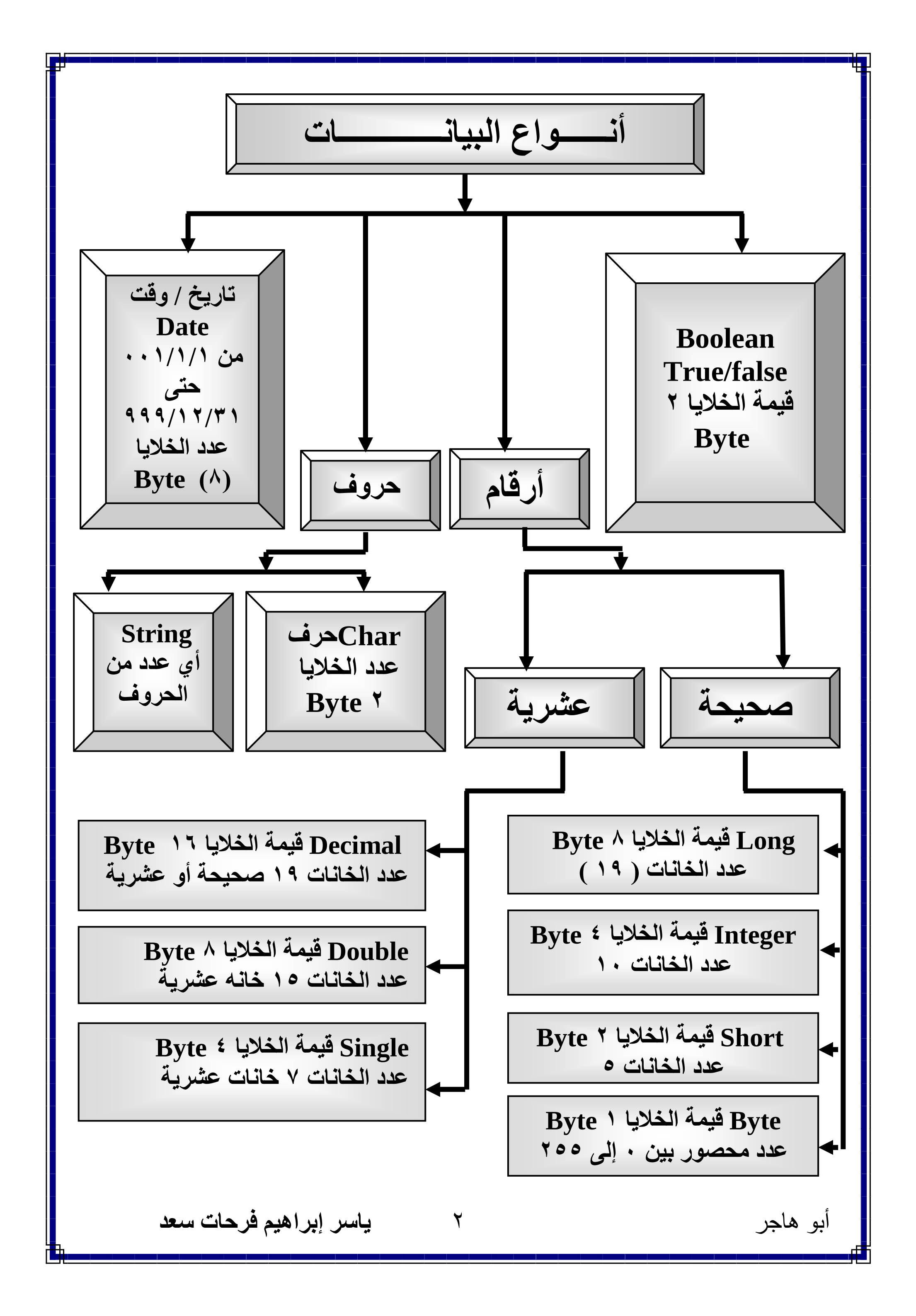
- تكلم عن ذاكرة الكمبيوتر الداخلية و مما تتكون ؟

١- أي بيانات تخزن يتم تخزينها في ذاكرة الكمبيوتر الداخلية
٢- ذاكرة الكمبيوتر تتكون من ملايين من الخلايا المتساوية قيمة كل منها واحد Byte .
٣- تحمل هذه الخلايا أرقاماً متسلسلة تبدأ من الصفر .
٤- يمثل عدد هذه الخلايا ذاكرة الكمبيوتر فكلما زادت زادت السعة التخزينية
٥- لغات البرمجة تسمح بإعطاء أسماء لهذه الخلايا أو مجموعه من الخلايا

- البيانات المتغيرة و البيانات الثابتة
١- إذا كانت القيم المخزنة تتغير أثناء تشغيل البرنامج فإن أسماء الخلايا التي تشغل هذه القيم يطلق عليها متغيرات Variables
٢- إذا كانت القيم المخزنة لا تتغير أثناء تشغيل البرنامج فإن أسماء الخلايا التي تشغل هذه القيم يطلق عليها متغيرات ثوابت Constants .

أنواع البيانات :-

يحدد نوع البيانات إذا كانت متغيراً أو ثابتاً نوع المعلومات التي سيتم تخزينها في الذاكرة المخصصة له حيث أن لكل نوع حجماً يختلف عن النوع الآخر .
فمثلاً حجم الأعداد العشرية يختلف عن حجم الأعداد الصحيحة .



قواعد تكوين الأسماء في لغة VB.Net .

١- يجب أن يبدأ بحرف من حروف الهجاء و يفضل أن يكون بالانجليزية.
٢- بعد الحرف الأول يمكن أن يأتي أي عدد من الحروف أو الأرقام .
٣- لا يحتوي الاسم على أي علامات خاصة (المسافة – النقطة – علامة الاستفهام – علامة التعجب ..الخ) إلا رمز Underscore مسموح به
٤- ألا يكون الاسم من الأسماء المحجوزة في لغة VB.Net مثل (Dim – Double – Print - end - close - ......الخ)
٥- ألا يزيد عن ٢٥٥ حرفاً و هو عدد كافي .
٦- غير مسموح بتكرار اسم المتغير في نفس النطاق و لكن ممكن تكراره في إجراء آخر .

قواعد يستحسن استخدامها :-

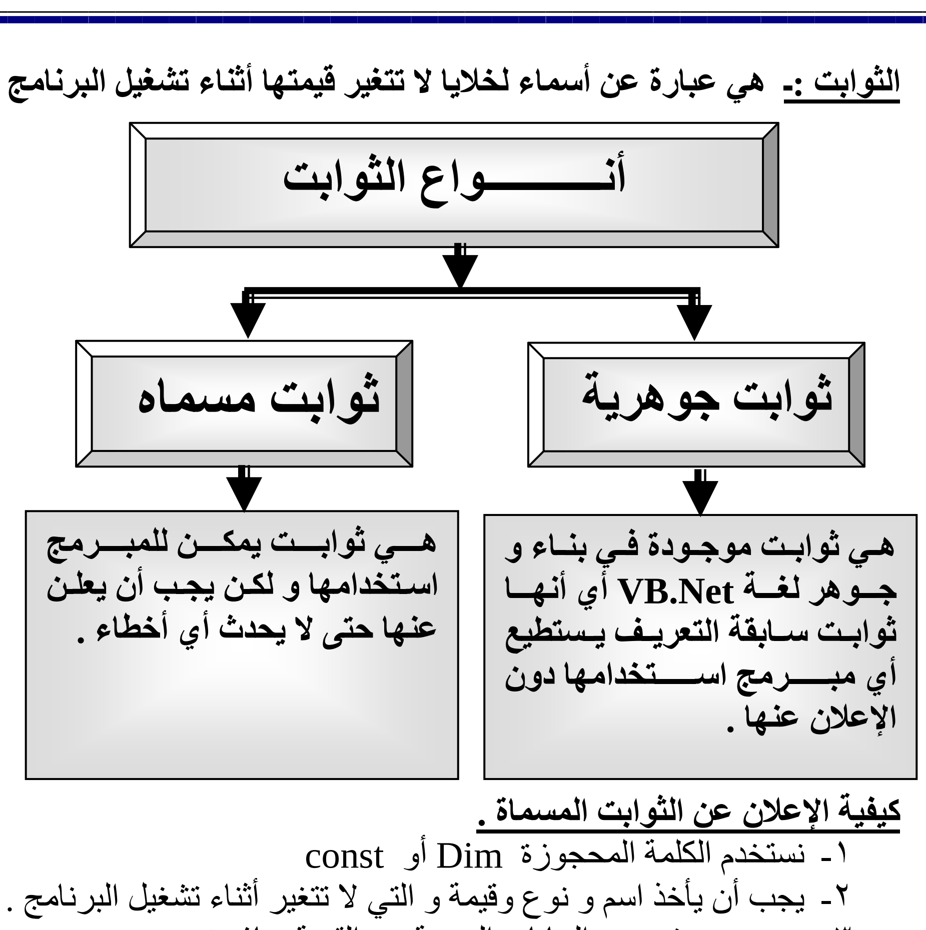
١- اختيار أسماء ذات معنى .
٢- يفضل أن يبدأ كل مقطع في اسم المتغير بحرف كبير (Capital)
٣- أن يبدأ اسم المتغير بثلاثة حروف تدل على نوع المتغير .

نشاط :-

ضع علامة (√) أمام الأسماء الصحيحة و علامة (×) أمام الأسماء الخطأ

اسم المتغير	√	×	سبب عد الصلاحية
1Name	....	....	....
Ahmed2001	....	....	....
Ibrahim_2010	....	....	....
Double	....	....	....
FirstName	....	....	....
Abo*Bakr	....	....	....
Data	....	....	....
Abo.bakr	....	....	....
Ahamad$2030	....	....	....
Mohamed 2010	....	....	....
Yasser!_Farahat	....	....	....
Print	....	....	....

الثوابت :- هي عبارة عن أسماء لخلايا لا تتغير قيمتها أثناء تشغيل البرنامج



كيفية الإعلان عن الثوابت المسماة .

١- نستخدم الكلمة المحجوزة Dim أو const
٢- يجب أن يأخذ اسم و نوع وقيمة و التي لا تتغير أثناء تشغيل البرنامج .
٣- يوضع حرف نوع البيانات العددية بعد القيمة مباشرة

ملاحظة :- إذا لم تستخدم حرف بعد القيمة فإن لغة VB.Net تعتبر الأعداد الصحيحة من النوع Integer و القيم العشرية من النوع double .

جدول اختصارات كل نوع من البيانات

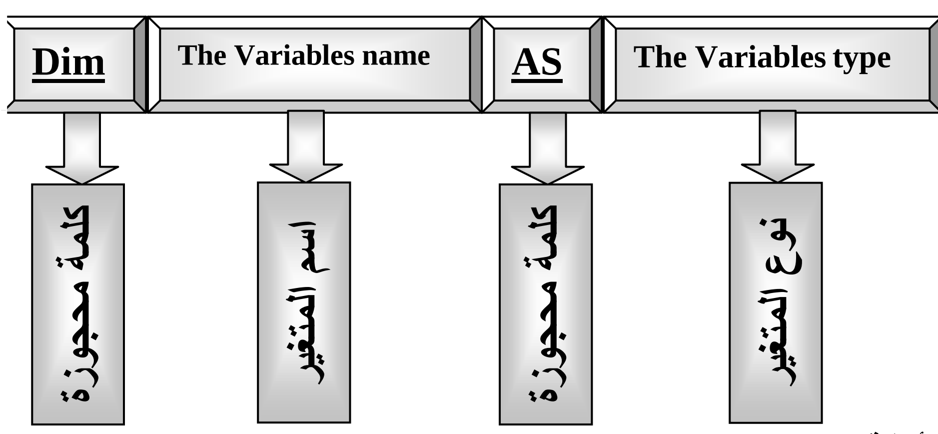
الحرف	النوع
S	Short
I	Integer
L	Long
D	Decimal
F	Single
R	Double

لماذا يفضل استخدام الثوابت عن استخدام قيم ؟
لأن استخدام الثوابت يتميز بـــ :-
١- يصبح البرنامج مفهوماً أكثر عند قراءته .
٢- لظروف معينه بعد تشغيل البرنامج فقد نضطر إلى تغير قيم بعض الثوابت .

المتغيرات

كيفية الإعلان عن المتغيرات .
١- تبدأ بكلمة Dim
٢- اسم المتغير
٣- كتابة كلمة AS
٤- نوع المتغير
٥- لك الحرية في إعطاء قيمة ابتدائية باستعمال علامة " = " ثم القيمة

القاعدة العامة للإعلان عن المتغير :-



أمثلة للإعلان عن المتغيرات :-

Dim UnitPrice AS Decimal
Dim UnitPrice AS Decimal = 120
Dom FirstName AS string



(١) عند إهمال القيمة الابتدائية فإن المتغير يأخذ قيم افتراضية
١- المتغيرات الرقمية تأخذ القيمة صفر .
٢- المتغيرات الحرفية تكون فارغة .

(٢) يمكن الإعلان عن أكثر من متغير باستخدام أمر Dim واحد
١- في حالة المتغيرات من نوع واحد

Dim S,Y,A,B As integer

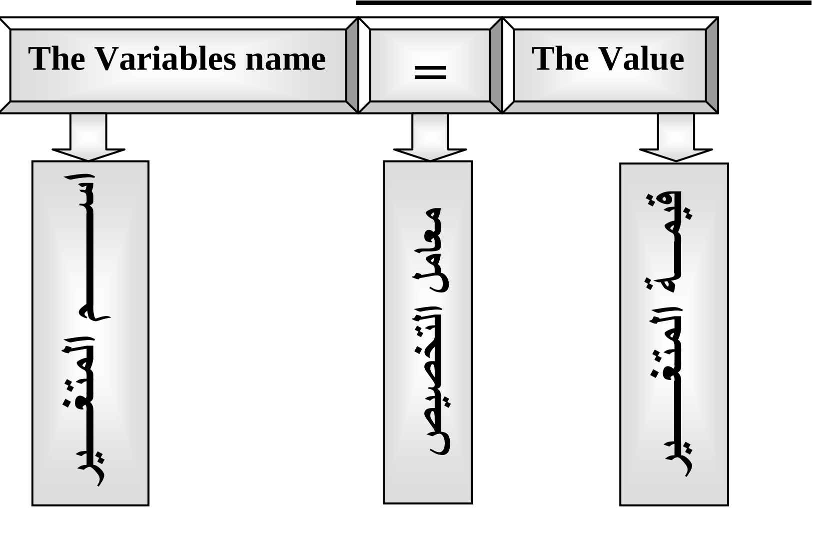
٢- في حالة المتغيرات ليست من نوع واحد .

Dim Tax As decimal = 1.5 , My Address As string = " Al Riad-KafrShih-Egypt"



دور المتغيرات هو تخزين البيانات بها و لكي نستطيع تخزين قيمة المتغيرات فإننا نستعمل أمر التخصيص .

القاعدة العامة لأمر التخصيص .



قواعد يجب مراعاتها لتنفيذ عملية التخزين :

١- اسم المتغير يجب أن يكون على يسار معامل التخصيص

٢- يتم تخزين القيم الرقمية في متغيرات رقمية و تخزن القيم الحرفية في المتغيرات الحرفية .

٣- القيم المراد تخزينها يمكن أن تأخذ أشكال متعددة مثل

أ- قيمه مجردة **مثل** $Y = 7$

ب- متغير آخر **مثل** $Y = H$ علما بأن $H = 9$

جـ - عمليه حسابية **مثل** $Y = H + 6 - 4$

٤- مفهوم معامل التخصيص " = " يختلف تماما عن مفهومها في المعادلات الرياضية

أمثلة :-

١- $Y + 4 = 10$ هذه معادلة صحيحة من الناحية الرياضية و لكنها غير صحيحة كعامل تخصيص .

٢- $Y = Y + 4$ غير صحيحة من الناحية الرياضية و لكنها صحيحة كمعامل تخصيص .

٥- المتغيرات التي على يمين معامل التخصيص لا يحدث بها أي تغير تظل قيمتها ثابتة بعد تنفيذ أمر التخصيص و الذي يتغير هـــو المتغير الذي على يسار معامل التخصيص .

س :- كيف يتم تنفيذ أمر التخصيص ؟

جـ :- لتنفيذ أمر التخصيص يقوم الكمبيوتر بعمل خطوتين هما :

أ- **تحديد القيمة الموجودة على يمين معامل التخصيص " = " كالآتي**

١- استبدال كل متغير أو ثابت بقيمته .

٢- إجراء العمليات الحسابية و استخراج النتيجة .

ب – **تخزين النتيجة في المتغير على شمال معامل التخصيص " = "**

مثال :

```
Quantity = 100
UnitPrice =5.5
Price = Quantity * UnitPrice
Price = 100 * 5.5
Price = 550
```


استخدام أمر التخصيص مع المتغيرات الحرفية (String)

كما هو الحال في استخدام أمر التخصيص في المتغيرات الحسابية يمكن استخدامه مع المتغيرات من النوع الحرفي و إجراء بعض العمليات عليها كربط أكثر من قيمة حرفية و معامل الربط هنا هو " & " .

مثال :-

```
Dim firstName as string
Dim LastName as string
Dim Message as String
FirstName ="Hagar"
LastName ="Yasser"
Message ="Welcome" &FirstName "  " &LastName
```

النتيجة ← Welcome Hagar Yasser

ملاحظة "" تعني مسافة طولها صفر أما "   " فإنها تعني قيمة حرفية طولها عدد المسافات

التحول بين الأنواع المختلفة للبيانات :

تختلف طريقة تخزين الأنواع المختلفة من البيانات في ذاكرة الحاسب كما يختلف حجم الذاكرة التي يتم التخزين فيها .

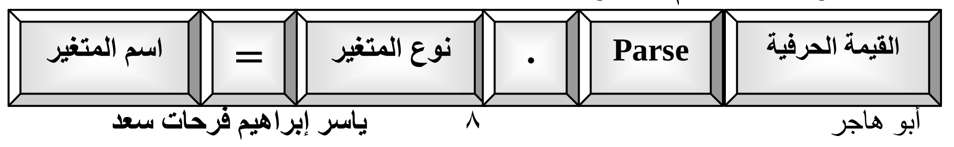
مثال :ـ تختلف طريقة تخزين القيمة الحرفية (١٠٠) عن طريقة تخزين القيمة الرقمية (١٠٠) حيث يتم تخزين القيمة الحرفية (١٠٠) في ٣ Byte أما القيمة الرقمية (١٠٠) يتم تخزينها في ١ Byte .

١ـ تحويل البيانات الحرفية إلى بيانات رقمية :

- أداة TextBox تستقبل البيانات على أساس أنها بيانات حرفية

- لكي يتم إجراء عمليات حسابية على المتغيرات أو الثوابت لابد من تحويلها من بيانات حرفية إلى بيانات رقمية .

- لتحويل البيانات الحرفية إلى بيانات رقمية نستخدم وسيلة Method اسمها Parse و الشكل العام لها هو .



مثال :-

Dim uprice as decimal

قم بتحويل قيمة السعر التي سوف يتم إدخالها في مربع الكتابة TextBox 1

جـ - Uprice = decimal.parse(me.textbox1.text)

ملاحظة يمكن عمل تعليقات أثناء كتابة الأكواد و لكن لابد أن تسبق بالعلامة (') بحيث لا تراها لغة VB.Net .

٢- تحويل البيانات الرقمية إلى بيانات حرفية :-

يمكن تحويل البيانات الرقمية إلى بيانات حرفية و ذلك باستخدام To string ( )



العمليات الحسابية :-

المعامل المؤثر	اسم العملية	مثال	ناتج العملية
+	الجمع	48 + 5	53
-	الطرح	48 – 5	43
*	الضرب	48 * 5	240
/	القسمة	48 / 5	9 3/5
\	قسمة أعداد صحيحة	48 \ 5	9
Mod	باقي القسمة	48 Mod 5	3
^	الأس	8 ^ 5	32768

أوامر Try /Catch :-

تستخدم هذه الأوامر لاكتشاف الأخطاء أثناء تشغيل البرنامج حيث يتم تنفيذ الأوامر التي تلي كلمة Try إذ لم يحدث أي أخطاء فإذا حدث خطأ من قبل مشغل البرنامج تخرج الرسالة التي تلي Catch و التي تفيد أن هناك خطأ من قبل مشغل البرنامج في إدخال البيانات يجب تصحيحها .

التحويل بين أنواع البيانات العددية :-

١- التحويل المباشر (التلقائي)

- يستخدم عند تحويل بيانات من نوع عددي ذا سعة تخزينية صغيرة إلى نوع عددي ذا سعة تخزينية أكبر . (يقصد بالسعة التخزينية عدد الخلايا التي تشغلها في الذاكرة)

أنواع التحويلات التي يمكن إجراؤها بالطريقة المباشرة :-

From	To
Byte	Short –Integer–Long–Single - Double – Decimal
Short	Integer–Long–Single - Double – Decimal
Integer	Long–Single - Double – Decimal
Long	Single - Double – Decimal
Decimal	Single - Double
Single	Double

ملحوظة :- التحويل المباشر لا يصلح في حالتين :-

١- تحويل البيانات من النوع Double إلى نوع آخر .

٢- تحويل البيانات التي تحتوي على كسور (Single أو Double) إلى النوع Decimal

في كلتا الحالتين نستخدم التحويل الغير مباشر

٢- التحويل الغير مباشر :-

- يتم استخدامه عندما نحول نوع ذو سعة تخزينية كبيرة إلى نوع أقل في السعة التخزينية .

- أي يستخدم بين البيانات العددية التي لا يمكن إجراء التحويل المباشر عليها مثل تحويل النوع Double إلى أي نوع آخر من أنواع البيانات .

- برنامج VB.Net يتيح Class لعملية التحويل الغير مباشر يسمى بـ Convert Class يحتوي على وسائل Methods تبدأ أسماؤها بـ to للتحويل بين الأنواع المختلفة .

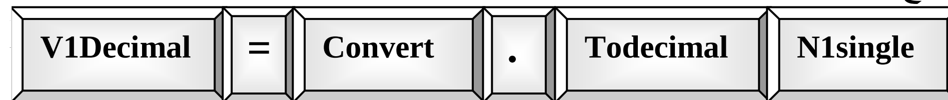
جدول يبين كيفية التحويل للبيانات

To convert to data type	Use Method (الطريقة المستخدمة)
Short	ToInt16
Integer	ToInt32
Long	ToInt64
Decimal	Todecimal
Single	Tosingle
Double	Todouble

مثال ١ :-

حول المتغير N1Single من النوع Single إلى المتغير V1Decimal من النوع Decimal .

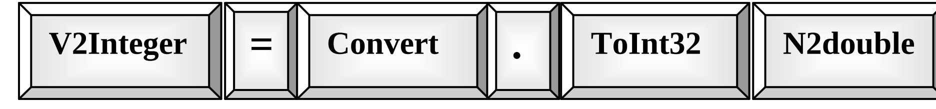
ج :-



مثال ٢ :-

حول المتغير N2Double من النوع Double إلى المتغير V2Integer من النوع Integer .

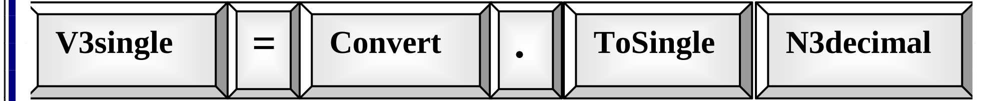
ج:-



مثال ٣ :-

حول المتغير N3decimal من النوع decimal إلى المتغير V3single من النوع Single .

ج:-



ترتيب تنفيذ العمليات الحسابية :-

يتم ترتيب تنفيذ العمليات الحسابية داخل VB.Net كالآتي :-

١- العمليات التي بين الأقواس (الداخلية أولا ثم الخارجية)

٢- عمليات الأس

٣- عمليات الضرب و القسمة (الأولوية من اليسار إلى واليمين)

٤- عمليات قسمة الأعداد الصحيحة

٥- عمليات حساب باقي القسمة

٦- عمليات الجمع و الطرح (من اليسار إلى اليمين)

مثال :- ما ناتج تنفيذ العمليات التالية :-

X = 2 , Y=3 , Z = 4

(1) X * (Y + 1)

(2) Y ^ X * X + Z * 3

ج:- **العملية الأولى و هي** X * (Y + 1)

١- نستبدل كل متغير بقيمته 2 * (3 + 1)

٢- نجري العمليات التي بداخل الأقواس و تكون النتيجة 2 * 4

٣- يتم تنفيذ عملية الضرب حتى تكون النتيجة 8

العملية الثانية و هي Y ^ X * X + Z * 3

١- استبدال كل متغير بقيمته 3^2*2+4*3

٢- يتم حساب الأس أولا ليعطي 9 * 2+ 4 * 3

٣- يتم تنفيذ عمليات الضرب من اليسار إلى اليمين 18 + 12

٤- يتم تنفيذ عملية الجمع حتى يكون الناتج 30

الفصل الثاني الشروط و القرارات

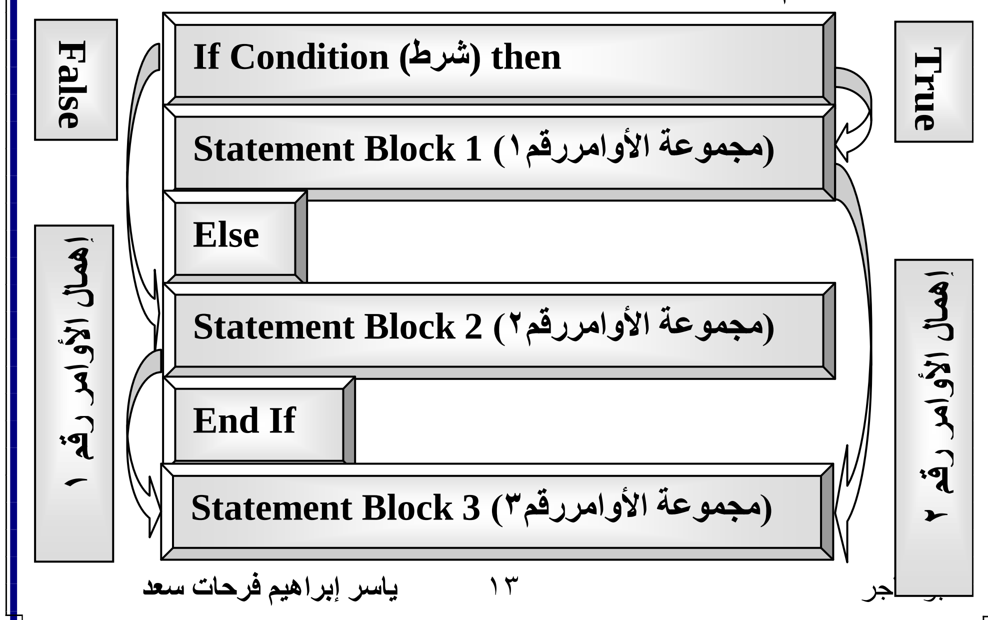
استخدام التعبيرات الشرطية :-

تعريف التعبير الشرطي : هو عبارة عن صياغة لعلاقة بين كميتين ممكن أن تكون متغير أو ثابت أو قيمة مجردة أو خاصية و التعبير الشرطي له قيمة تتوقف عليه صحة العلاقة أو عدم صحتها فالحالة الصحيحة تسمى True و الحالة الخاطئة تسمى False .

جدول يبين معاملات المقارنة :-

معامل المقارنة	العلاقة المطلوب اختبارها
>	أكبر من
<	أصغر من
=	تساوي
<>	لا تساوي
>=	أكبر من أو تساوي
<=	أصغر من أو تساوي

قاعدة استخدام الأمر If ....then.........Else



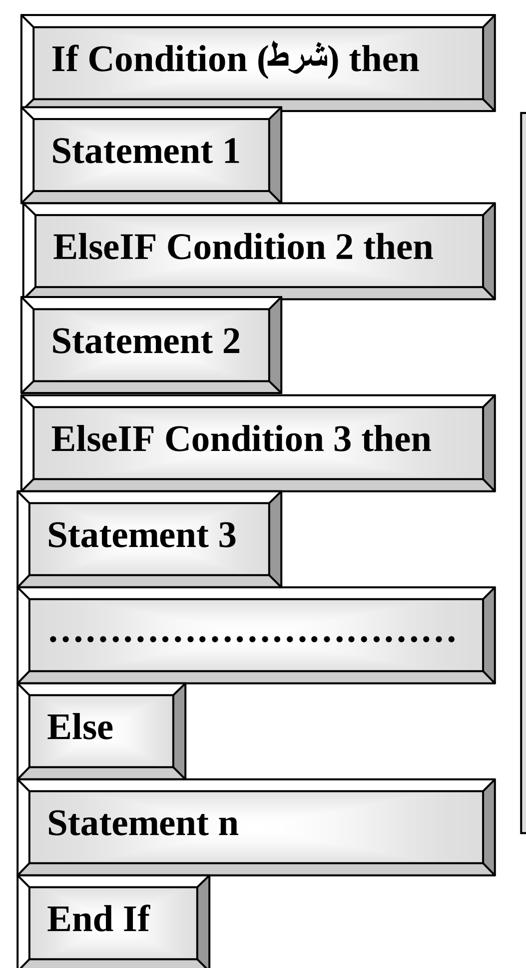
إذا كان الشرط True

يقوم البرنامج بتنفيذ الأوامر ١ الموجودة بعد كلمة Then ثم يذهب إلى مجموعة الأوامر ٣ الموجودة بعد End If مع إهمال مجموعة الأوامر ٢.

إذا كان الشرط False

يتم تنفيذ مجموعة الأوامر ٢ و التي بعد كلمة Else ثم يذهب البرنامج إلى تنفيذ مجموعة الأوامر ٣ الموجودة بعد End If مع إهمال مجموعة الأوامر ١

الشكل العام للأمر If.......then.........Else



عند تنفيذ هذا الأمر يتم اختبار قيمة كل شرط من أعلى إلى أسفل و عندما يقابل أول شرط صحيح أو True يقوم بتنفيذ مجموعة الأوامر الخاصة بهذا الشرط و يتوقف البرنامج فإذا لم يجد أي قيم صحيحة فإنه يذهب إلى مجموعة الأوامر التي تلي كلمة Else فإذا لم توجد Else فلا يتم تنفيذ أي أمر و يذهب البرنامج إلى أول أمر بعد End If .

<u>قاعدة استخدام الأمر If ..... then البسيط :-</u>

```
If Condition then
Statement block 1
End If
Statement block 2
```

١- إذا كان الشرط **True** سيقوم البرنامج بتنفيذ مجموعة الأوامر رقم ١ ثم يذهب لتنفيذ مجموعة الأوامر رقم ٢ .

٢- إذا كان الشرط **False** فإن البرنامج سوف يذهب مباشرة لتنفيذ مجموعة الأوامر رقم ٢ التي تلي **End If** .

مثال :-

```
If Age > 20 then
Mgbox(" أنت رجل كبير ")
End If
Msgbox(" أهلا بك مع عالم البرمجيات ")
```

في هذا المثال السابق :-

١- <u>في حالة تحقق الشرط :</u>

تخرج رسالة أنت رجب كبير ثم تخرج رسالة أهلا بك مع عالم البرمجيات .

٢- <u>في حالة عدم تحقق الشرط :</u>

تخرج رسالة واجدة و هي أهلا بك مع عالم البرمجيات .

س:- ما الفرق بين IF.....then.....Else و If.......then ؟

ج :-

١- الأمر If ...then ..... Else يتم استخدامه إذا كان هناك مفاضلة بين بديلين فإذ كان ناتج الشرط True يتم تنفيذ البديل الأول و إذا كان ناتج الشرط False يتم تنفيذ البديل الثاني .

٢- الأمر If....then يتم استخدامه في حالة ما إذا كان هناك بديل واحد فإذا كان ناتج الشرط True يتم تنفيذ الأمر الموجود بعد Then و إذا كان ناتج الشرط False لا يتم عمل شيء .

<u>الشروط المركبة :-</u>

تستخدم في حالة ما إذا كنا سنستخدم قراراً معيناً بناءاً على تحقق شرطين و ليس شرط واحد و يمكننا عمل الشرط المركب بالربط بين أكثر من شرط بأحد المعاملات التي تسمى المعاملات المنطقية و هي .

المعامل	المعنى
OR	إذا كان الشرطين صحيحين أو أحدهما صحيح و الآخر غير صحيح تكون النتيجة True .
And	إذا كان كل من الشرطين صحيح تكون النتيجة True غير ذلك تكون النتيجة False .
Not	إذا كان الشرط صحيحاً تكون النتيجة False و العكس إذا كان الشرط خطأ تكون النتيجة True بمعنى أنها تعكس الشرط .

<u>مقارنة القيم و المتغيرات الحرفية (String) :-</u>

يتم مقارنة المتغيرات الحرفية عن طريق تحديد رتبة كل رمز طبقاً لطريقة تخزين الرموز في ذاكرة الحاسب و هذه الطريقة تسمى ASCII Code .

ASCII = American Standard Code For Information Interchange

و هي تعطي كل حرف أو رقم أو رمز خاص رتبة معينه و هذه الرتبة هي رقم ثابت لا يتغير .

الأداة RadioButton :-

ذكر أنثى

و هي تتيح للمستخدم مجموعه من البدائل لاختيار بديل واحد فقط .

خصائص أداة RadioButton

Property	Action
Name	تحديد اسم الأداة و يفضل أن يبدأ بـــ Rad
Backcolor	تحديد لون خلفية الأداة
Appearance	تحتوي على اختيارين هما Normal و Button
Checked	تحديد ما إذا كانت الأداة تم اختيارها أم لا
Enabled	تحديد ما إذا كان المستخدم يستطيع التعامل مع الأداة أم لا
Forecolor	تحدبد لون الكتابة بداخل الأداة
Height	تحديد ارتفاع الأداة
Image	تحديد ما إذا كانت الأداة سوف تحتوي على صورة أم لا
Text	تحديد الكتابة التي سوف تظهر بداخل الأداة
Visible	تحديد ما إذا كانت الأداة مرئية أم لا
Width	تحديد عرض الأداة

اللغات التي تتحدثها
اللغة العربية
اللغة الانجليزية
اللغة الالمانية
اللغة الفرنسية

الأداة CheckBox :-

تتيح للمستخدم مجموعة من البدائل ممكن اختيار أكثر من بديل في نفس الوقت بعكس أداة RadioButton .

خصائص أداة CheckBox :-

Property	Action
Name	تحديد اسم الأداة و يفضل أن يبدأ بـ Chk
Backcolor	تحديد لون خلفية الأداة
Appearance	تحتوي على اختيارين هما Normal و Button
Checked	تحديد ما إذا كانت الأداة تم اختيارها أم لا
Enabled	تحديد ما إذا كان المستخدم يستطيع التعامل مع الأداة أم لا
Forecolor	تحديد لون الكتابة بداخل الأداة
Height	تحديد ارتفاع الأداة
Image	تحديد ما إذا كانت الأداة سوف تحتوي على صورة أم لا
Text	تحديد الكتابة التي سوف تظهر بداخل الأداة
Visible	تحديد ما إذا كانت الأداة مرئية أم لا
Width	تحديد عرض الأداة

الأداة GroupBox :-

هي أداة تستخدم لتنظيم الأدوات على النموذج على هيئة مجموعات حيث يستطيع مشغل البرنامج أن يختار من كل مجموعه ما يشاء .

خصائص أداة GroupBox :-

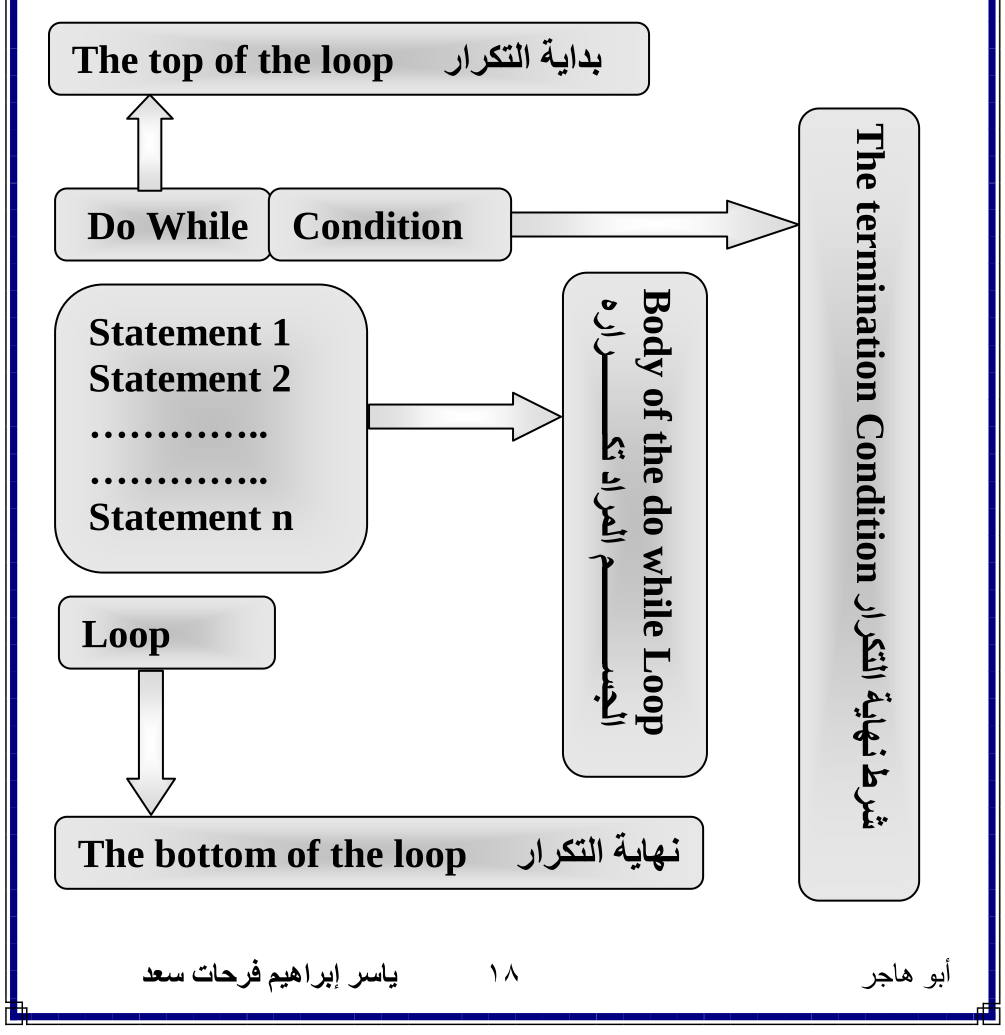
Property	Action
Name	تحديد اسم الأداة و يفضل أن يبدأ بـ grp
Enabled	تحديد ما إذا كان المستخدم يستطيع التعامل مع الأدوات التي بداخلها أم لا يستطيع .
Flat Style	تحديد شكل الأداة system – Flat – Popup - Standard
visible	تحديد ما إذا كانت الأداة و ما بداخلها من أدوات مرئية أم لا

الفصل الثالث أوامر التكرار

جميع الأوامر في اللغة VB.Net يتم تنفيذها من أعلى إلى أسفل فيما عدا الأمرين If و Try / Catch حيث أنهما يتجاهلان تنفيذ بعض الأوامر .
أما أوامر التكرار هي أوامر تتحكم في إعادة تنفيذ عمل معين أو مجموعة من الأوامر حتى يتحقق شرط معين .

الأمر Do While......loop :-

القاعدة العامة لكتابة الأمر Do While ......Loop .



كيفية عمل الأمر Do While .....loop : يقوم البرنامج باختبار الشرط

١- **إذا كان الشرط True** يقوم بتنفيذ الأوامر التي تلي الشرط ثم يقوم البرنامج باختبار الشرط مره ثانية فإذا كان True يقوم بإعادة تنفيذ الأوامر التي تلي الشرط ثم يقوم البرنامج باختبار الشرط مرة ثالثة و هكذا حتى يصبح الشرط False فيقف البرنامج كما هو واضح من الشكل التالي .

٢ Do While Condition → Condition is True

Statement 1
Statement 2
…………..
…………..
Statement n

١ الأوامر التي تنفذ

Loop

يتم اختبار الشرط مرة ثانية

Statement X

٢- **إذا كان الشرط خطأ False** يذهب البرنامج إلى مجموعة الأوامر التي تلي كلمة Loop و يتجاهل الأوامر التي الشرط كما يتضح من الشكل التالي .

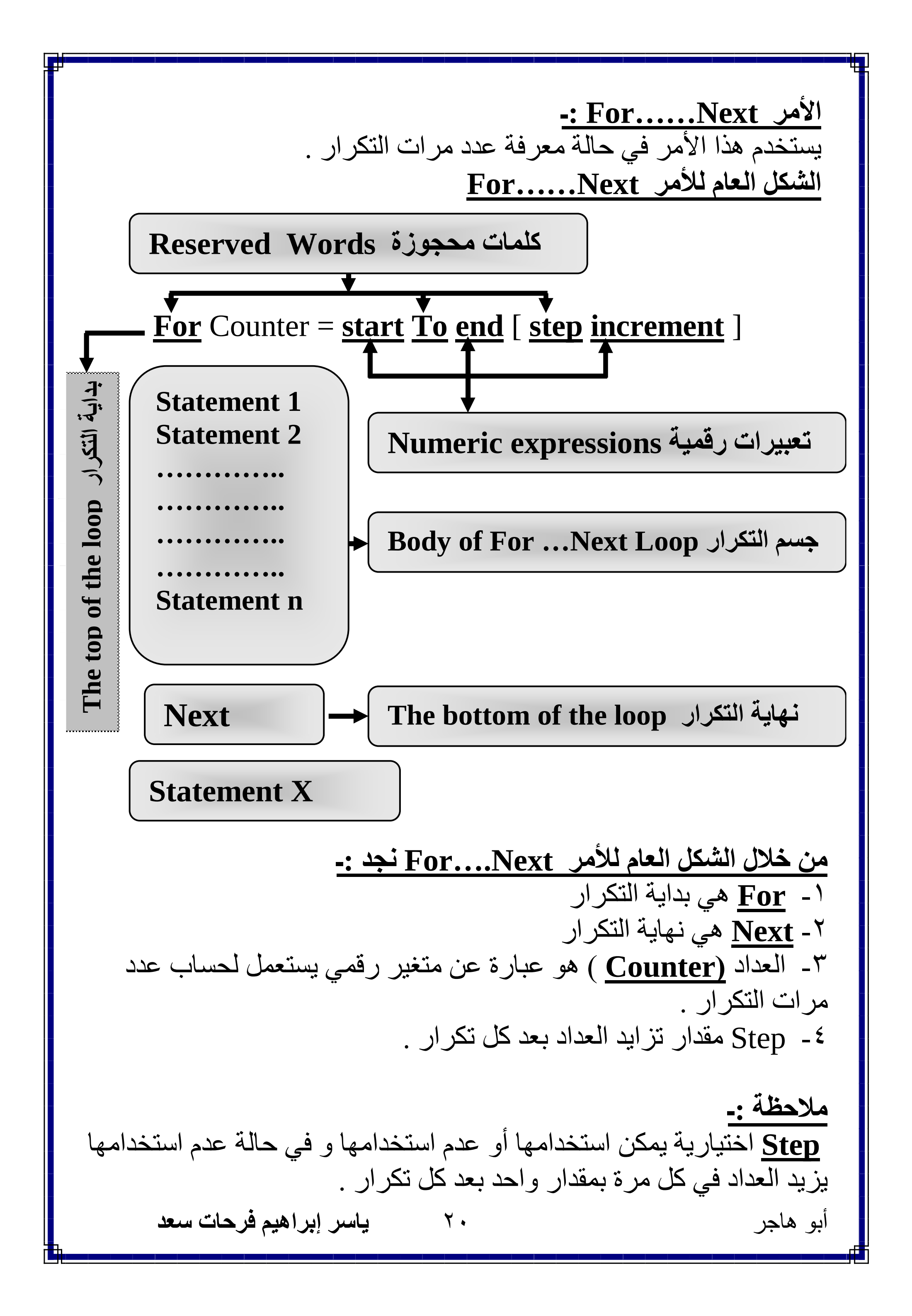
Do While Condition → Condition is False

Statement 1
Statement 2
…………..
Statement n
→ الجسم المراد تكراره

Loop

Statement X

الأوامر الذي ينفذ



الأمر For……Next :-

يستخدم هذا الأمر في حالة معرفة عدد مرات التكرار .

الشكل العام للأمر For……Next

من خلال الشكل العام للأمر For….Next نجد :-

١- **For** هي بداية التكرار

٢- **Next** هي نهاية التكرار

٣- العداد (**Counter**) هو عبارة عن متغير رقمي يستعمل لحساب عدد مرات التكرار .

٤- Step مقدار تزايد العداد بعد كل تكرار .

ملاحظة :-

Step اختيارية يمكن استخدامها أو عدم استخدامها و في حالة عدم استخدامها يزيد العداد في كل مرة بمقدار واحد بعد كل تكرار .

كيفية تنفيذ الأمر For….Next :-

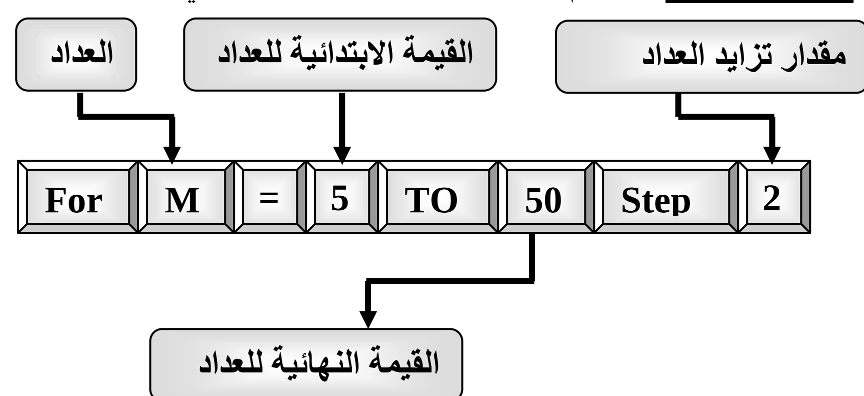
١- يتم تخزين القيمة الرقمية (Start) في العداد .

٢- تتم مقارنة القيمة المخزنة في العداد بالقيمة End حيث :-

أ- إذا كانت قيم العداد أكبر من القيمة End ينتهي التكرار و يذهب البرنامج إلى أول أمر بعد كلمة Next .

ب – إذا كانت قيم العداد أقل من قيمة End أو تساويه يتم تنفيذ التكرار الموجود بين For ، Next من أعلى إلى أسفل .

مثال توضيحي:- غير قيم العداد من ٥ إلى ٥٠ بزيادة ٢ في كل مرة .



أمثلة غير محلولة :-

١- غير قيم العداد S من ١ إلى ٢٠٠ بزيادة ٢ في كل مره .

.... أو ....

٢- غير قيم العداد H من ١٠٠ إلى ٢ بزيادة (- ٢) في كل مره

....

٣- غير قيم العداد N بالترتيب الآتي (19 , 17 , 15 , 13 , 11)

....

٤- غير قيم العداد Y من ١ إلى ٢٠٠

.... أو ....

٥- غير قيم العداد R بالترتيب الآتي (10 , 12 , 14 , 16 , 18 , 20)

....

الأداة ListBox :-

١- تعد هذه الأداة من الأدوات التي ترتبط غالباً بالتكرار .

٢- تتيح هذه الأداة مجموعة من العناصر نختار منها ما نشاء .

٣- تظهر هذه الأداة على هيئة مستطيل يعرض بداخله مجموعه من الصفوف.

خصائص أداة ListBox :-

Property	Action
Name	تحديد اسم الأداة
Horizontal Scrollbar	تحديد ما إذا كان هناك شريط منزلق أفقي أم لا
Item	معرفة العناصر الموجودة داخل الأداة و هذه الأداة تعتبر كائن له خصائص و وسائل .
Selected Item	معرفة العناصر الواقع عليها الاختيار
Sorted	تحديد ما إذا كانت العناصر داخل الأداة مرتبة أم لا و تأخذ قيمة True / False .

طرق Methods الخاصية Item للأداة ListBox

Method (الوسيلة)	Behavior (العمل)
Add (text)	**إضافة النص الذي بين القوسين أخر عناصر القائمة**
Count	تحديد عدد عناصر القائمة
Insert (Index , text)	**إضافة العنصر المحدد (text) في المكان المحدد (Index)**
Remove (text)	البحث عن العنصر (text) وإذا وجد تم إزالته
RemoveAt(index)	**إزالة العنصر المحدد (index) مع تحريك كل العناصر الموجود أسفل منه إلى أعلى خطوة واحدة**
Clear ( )	

ملاحظة :-

Index لـ Item في أداة ListBox تأخذ أرقام من الصفرو حتى (Coun-1) حيث أن count عدد العناصر الموجودة في ListBox .

الفصل الرابع الميقاتي

- الحدث Event يقع عندما يقوم المستخدم بالضغط على زر معين
- قد نحتاج حدوث الحدث Event دون تدخل مشغل البرنامج
- أداة Timer تستخدم لهذا الغرض و هي عبارة عن ساعة غير مرئية تتيح التعامل مع ساعة النظام من داخل البرامج التي تقوم باستخدامها .

استعمال الكائن Timer :-

١- ينشط الــ Timer على فترات زمنية متساوية تسمى هذه الفترات Intervals فيقع الحدث Tick و بالتالي يقوم بتنفيذ مجموعة من الأوامر .

٢- يمكن التحكم في الفترات التي ينشط فيها Timer و ذلك عن طريق الخاصية Intervals .

٣- قيمة Intervals تتراوح ما بين الصفر و ٦٥٥٣٥ مللي ثانية .

٤- الثانية الواحدة تساوي ١٠٠٠ مللي ثانية .

مثال :- إذا كان لديك Intervals قيمتها ١٧٠٠ مللي ثانية فما قيمتها بالثانية ؟

ج :- Intervals بالثانية = ١٧٠٠ ÷ ١٠٠٠ = ١,٧ ثانية .

٥- إذا أرت استخدام الــ Timer فيجب عليك أن تجعل قيمة الخاصية Enabled تساوي True لأن القيمة الافتراضية لها هي False .

٦- إذا أردنا إيقاف الـ Timer نجعل قيمة خاصية Enabled = False .

التكوين Date Time :-

عند الاعلان عن متغير من النوع Date يكون لدينا تكوين يسمى Date Time هذا التكوين له خصائص ووسائل كما تضح من الجداول الآتية .

جدول خصائص التكوين Date Time

Property	الغرض منها
Now	استرجاع التاريخ و الوقت الآن
Date	إدراج التاريخ
Day	إدراج رقم اليوم خلال الشهر
DayOfYear	إدراج رقم اليوم خلال السنة
Hour	إدراج الساعة
Minute	إدراج الدقيقة
Second	إدراج الثانية
Month	إدراج اسم الشهر حيث أن ١ يعني شهر يناير

جدول وسائل التكوين Date Time

Method	الغرض
AddDays	إنشاء تاريخ جديد متقدم (متأخر) عدد من الأيام
AddHours	إنشاء تاريخ جديد متقدم (متأخر) عدد من الساعات
AddMinutes	إنشاء تاريخ جديد متقدم (متأخر) عدد من الدقائق